

Trabalho 3

Introdução ao Processamento Digital de Imagem (MC920/MO443)
Professor: Hélio Pedrini

Vinicius Couto Espindola | RA: 188115

20 de Maio de 2020

1 Introdução

Parte da extração de significado das imagens refere-se as formas que tal imagem contém, por exemplo, uma foto de um texto apresenta diversas formas que são letras, palavras, linhas e parágrafos. Para trabalhar com tais dados vinculados à estas formas, podemos utilizar operadores morfológicos derivados da matemática que possibilitam transformações sobre estas formas. Este trabalho tem como objetivo mostrar como podemos utilizar o conceito de operadores morfológicos para extrair significado das formas da imagem, visando segmentar o que são linhas e o que são palavras dentro do conjunto de formas presentes em uma imagem.

2 Execução

O programa requer as seguintes bibliotecas: *OpenCV* e *NumPy*. Sua execução pode tanto exibir a imagem de saída, quanto salvar o resultado em um arquivo. Nota-se que todos os resultados serão salvos no formato *PBM*, no entanto, para facilitar a visualização do processo, há resultados intermediários salvos como *PNG*. Os resultados produzidos encontram-se na pasta `./outputs`. As imagens de entrada utilizadas estão na pasta `./inputs`. O programa trata apenas entradas do tipo *PBM*, mas, caso necessário, este pode converter uma imagem *PNG* para *PBM* usando a flag `-b`.

`assignment_4.py [-h] [-b threshold] [-s] [-d] [-v] [-vv] [-o output] [-l] [-a]input`

Nota: A execução padrão (definindo apenas a imagem de entrada) ira imprimir a segmentação de palavras obtidas no formato *PNG*. Para salvá-las em *PBM* use a flag `-o`.

input: *Path* da imagem de entrada a ser processada.

-b threshold: Caso esta flag seja setada, a imagem de entrada será convertida para um *PBM* de forma que pixels com intensidade menores e iguais ao *threshold* são truncados à zero e o restante para 255.

Nota: se a flag `-b` for usada, todas as outras flags são desconsideradas. Para testar com imagens *PNG*, primeiro converta a imagem, depois aplique a segmentação.

-s: Exibe o resultado da segmentação na tela no formato *PNG* com coloração.

-d: Flag para depuração. Esta ajuda à identificar se os parâmetros selecionados para a identificação de texto são adequados.

-v: Modo com verbosidade. Este imprime, além do resultado final, as componentes conexas da imagem que representam linhas e as que representam palavras.

-vv: Modo com dupla verbosidade. Este imprime os resultados de todos os passos efetuados nas operações morfológicas.

-o: Caso não seja definido um *output-file*, a imagem será exibida na tela. Se for definido, a imagem de saída será salva apenas, a não ser que a flag `-s` seja utilizada. **-l:** Invés de segmentar as palavras, que é o caso padrão, segmenta apenas as linhas.

-a: Segmenta tanto as linhas quanto as palavras dada uma imagem.

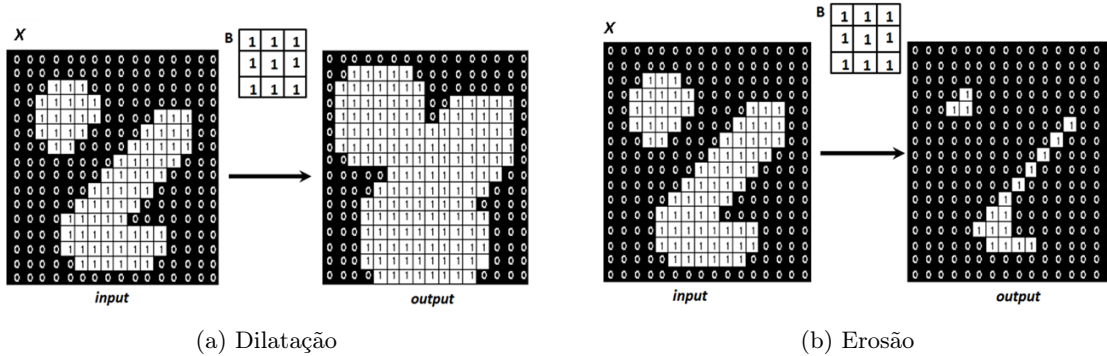


Figure 1: Operações morfológicas básicas em uma imagem X usando um $kernel$ B . ([source](#))

3 Soluções

Operações morfológicas podem ser facilmente realizadas com as funções já implementadas no *OpenCV*. Neste trabalho veremos duas operações morfológicas básicas e uma composta que derivam das básicas.

As funções do *OpenCV* consideram uma imagem binária onde 1 é branco e 0 é preto (ou 255 e 0), de forma que os pixels brancos fazem parte do conjunto sobre o qual o elemento estruturante aplica as operações de conjuntos e os pixels pretos são ignorados.

3.1 Conceitos Base

Duas operações morfológicas clássicas são a dilatação e a erosão. A dilatação tem o efeito de alargar/juntar as formas presentes na imagem, enquanto a erosão tem o efeito de afinar/quebrar as formas. Podemos visualizar o efeitos destas operações utilizando um $kernel$ 3×3 na **Figura 1**

A partir destas operações básicas, podemos definir a operação de fechamento: ao aplicarmos a dilatação seguida da erosão com um $kernel$ adequado, podemos fechar lacunas entre as formas das imagens, assim como eliminar ruídos (ou buracos). Na etapa da dilatação, temos que as lacunas e buracos da imagem são tampados, mas a forma também sofre um alargamento de tamanho, este alargamento é eliminado pela operação de erosão aplicada em seguida. O processo de fechamento esta ilustrado na **Figura 2**

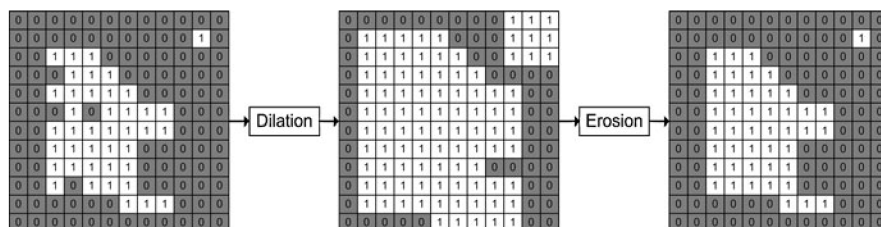


Figure 2: Resultado do fechamento em uma imagem usando um $kernel$ 3×3 cheio. ([source](#))

3.2 Segmentação Morfológica

A operação de fechamento, quando combinada com o $kernel$ adequado, pode gerar efeitos úteis sobre uma imagem. Para exemplificar como utilizar estas operações para segmentação de texto, aplicaremos o fechamento da **Figura 3** com um $kernel$ horizontal de 1×100 e um vertical de 200×1 . Após os fechamentos, faremos a intersecção e refinamento dos resultados. O refinamento consiste em aplicar um fechamento com um $kernel$ pequeno de 1×30 , o qual serve para eliminar o ruído dos objetos resultantes da intersecção. A imagem em questão trata-se de uma página de um livro, mas, como o *openCV* considera que os pixels brancos são os objetos da imagem, a **Figura 3** é a página negada (branco e preto invertidos), assim o texto da imagem será considerado como um objeto.

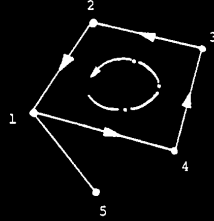
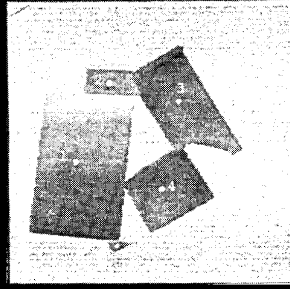


Fig. 4. Range image of an AMBIGUOUS scene and the corresponding graph.

sensory feedback is carried out in a local reflexive mode rather than in a planned mode with one exception, that is, when a pathological state is detected.

3) *States*: This is a finite set of states describing the environment of the Turing machine as perceived by the sensors. If new sensors are added, the set of states is partitioned to describe the scene as perceived by the additional sensors. For example, if a sensor capable of determining the "touch" relations of objects in the scene is added, then the set of five states, can be partitioned (a finer partition) to describe both the "touch" and "on-top-of" relations. The states of the machine are:

Empty	If there are no vertices in the diagram, i.e., an empty diagram.
Dispersed	If there no edges in the diagram, i.e., a null diagram (Fig. 2).
Overlapped	If there are at least two vertices connected with an edge (Fig. 3).
Ambiguous	If there is one or more directed cycles in the diagram (Fig. 4).
Unstable	This category is not tested by the analysis of the graph but through analysis of the contact point/line of the object with the support plane. If this contact is a point or a line, it is classified as unstable. See Fig. 5.

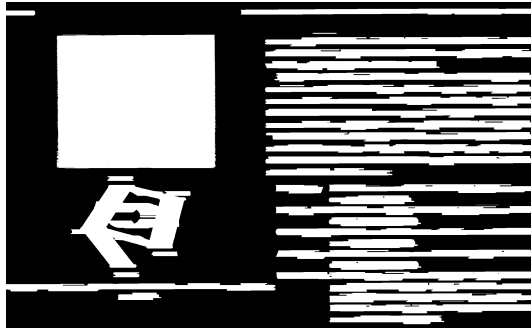
Figure 3: Imagem PBM de entrada negada.

Observa-se na **Figura 4** que os *kernels* utilizados conseguem transformar os objetos da imagem em formas conexas. O fechamento vertical (*c* e *d*) é redundante em relação ao horizontal (*a* e *b*) neste caso, todavia, caso a imagem estivesse mais próxima ao bloco de texto da direita, o fechamento vertical seria essencial para isolar o texto e a imagem. Percebe-se que a intersecção em **e** na **Figura 4** cria uma série de traços verticais que tornam as linhas em formas desconexas, o refinamento então cuida deste ruído para que obtenhamos o resultado final em **f**. O resultado final trata-se da segmentação da imagem de forma que cada objeto (blocos brancos) é uma região conexa, podemos, então, isolar as componentes conexas utilizando a função *connectedComponentsWithStats* do *OpenCV*: esta retorna retângulos que englobam cada uma das componentes conexas.

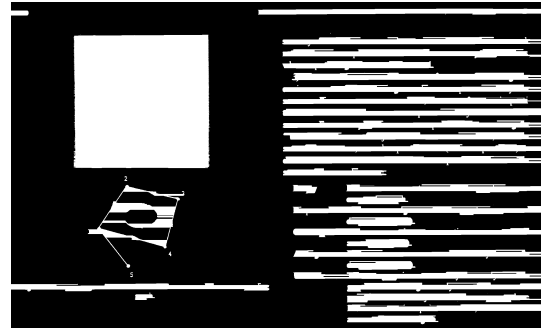
3.3 Detectando Linhas

Apesar de conseguirmos identificar os objetos da imagem, ainda não sabemos distinguir o que é cada objeto. Podemos, então, tentar isolar o que é texto de acordo com as características do objeto que representa texto. Consegue-se extrair duas informações úteis dos retângulos que englobam as componentes conexas: a razão de pixels pretos em relação ao total de pixels do bloco (BTR) e a razão entre o número de transições preto-branco/branco-preto em relação ao número total de pixels pretos (MTR).

Podemos, então, iterar por todos os blocos que envolvem componentes conexas e calcular estes dois valores. Experimentalmente, determina-se que as linhas tem BTR entre 0.5 e 0.9 e MTR entre 0 e 0.08, ou seja, para todo bloco os quais apresentam as duas razões mencionadas entre estes intervalos, consideraremos que a componente envolvida pelo bloco em questão trata-se de uma linha. O resultado deste processo esta represetado na **Figura 5**. Dado que sabemos quantas componentes representam linhas, podemos associar este valor ao número de linhas. Na **Figura 5** foram reconhecidas 34 linhas.



(a) Dilatação Horizontal



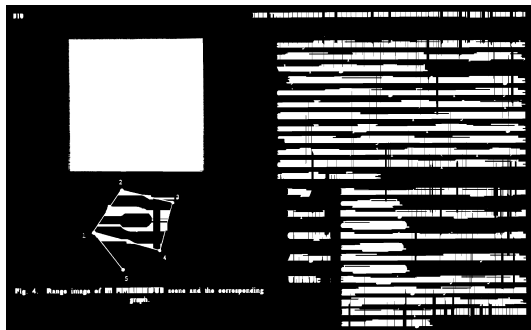
(b) Erosão Horizontal



(c) Dilatação Vertical



(d) Erosão Vertical



(e) Intersecção



(f) Refinamento

Figure 4: Segmentação de linhas de texto utilizando operadores morfológicos.

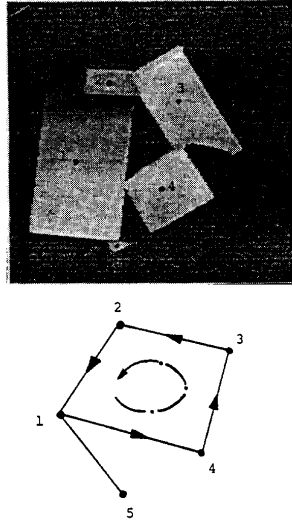


Fig. 4. Range image of an AMBIGUOUS scene and the corresponding graph.

sensory feedback is carried out in a local reflexive mode rather than in a planned mode with one exception, that is, when a pathological state is detected.

3) *States*: This is a finite set of states describing the environment of the Turing machine as perceived by the sensors. If new sensors are added, the set of states is partitioned to describe the scene as perceived by the additional sensors. For example, if a sensor capable of determining the "touch" relations of objects in the scene is added, then the set of five states, can be partitioned (a finer partition) to describe both the "touch" and "on-top-of" relations. The states of the machine are:

Empty	if there are no vertices in the diagraph, i.e., an empty diagraph.
Dispersed	if there no edges in the diagraph, i.e., a null diagraph (Fig. 2).
Overlapped	if there are at least two vertices connected with an edge (Fig. 3).
Ambiguous	if there is one or more directed cycles in the diagraph (Fig. 4).
Unstable	This category is not tested by the analysis of the graph but through analysis of the contact point/line of the object with the support plane. If this contact is a point or a line, it is classified as unstable. See Fig. 5.

Figure 5: Demarcação (azul) dos blocos os quais englobam uma componente correspondente à uma linha.

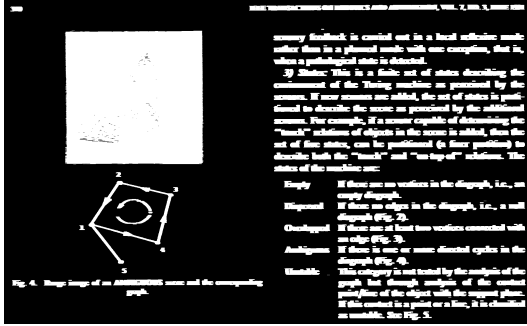
3.4 Detectando Palavras

Da mesma forma que fizemos a identificação das linhas, podemos realizar a identificação das palavras, no entanto devemos mudar os seguintes parâmetros: tamanhos dos *kernels* utilizados nas operações de fechamento e os intervalos dos atributos BTR e MTR que representam agora palavras e não linhas. Podemos, então, apenas regular os parâmetros para que estes segmentem palavras invés de linhas.

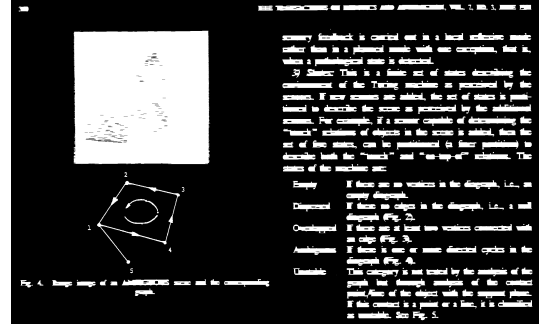
Para a segmentação de palavras, foi utilizado os seguintes parâmetros:

- *Kernel* de 1X12 para o fecho horizontal.
- *Kernel* de 4X1 para o fecho vertical.
- *Kernel* de 8X12 para o fecho de refinamento.
- Intervalos de [0.35, 0.95] e [0, 0.2] para BTR e MBR respectivamente.

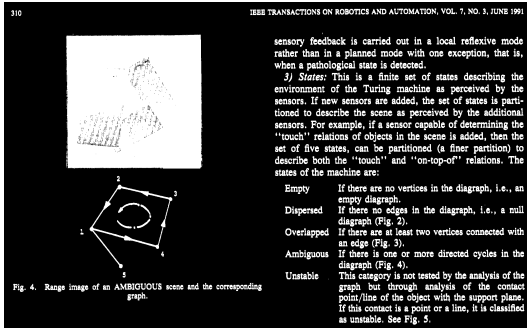
Dados estes parâmetros, temos os resultados intermediários das operações ilustrados na **Figura 6**, e a segmentação das palavras da imagem de entrada na **Figura 7**. Para esta figura, tivemos uma contagem de 237 palavras.



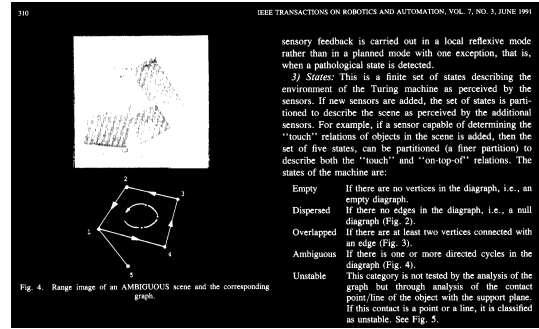
(a) Dilatação Horizontal



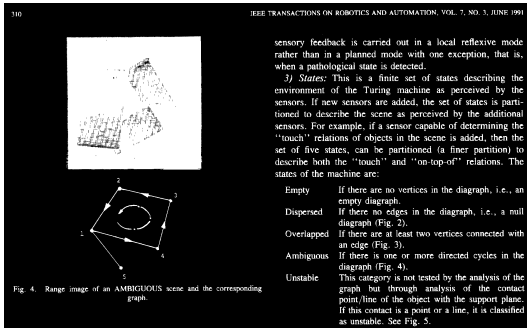
(b) Erosão Horizontal



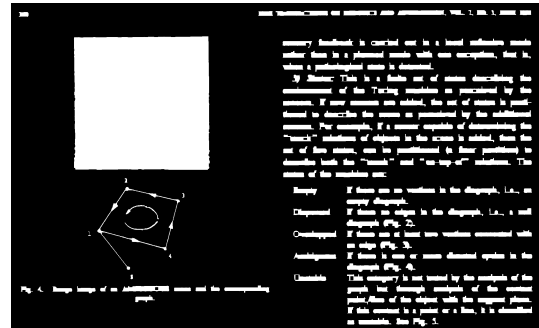
(c) Dilatação Vertical



(d) Erosão Vertical

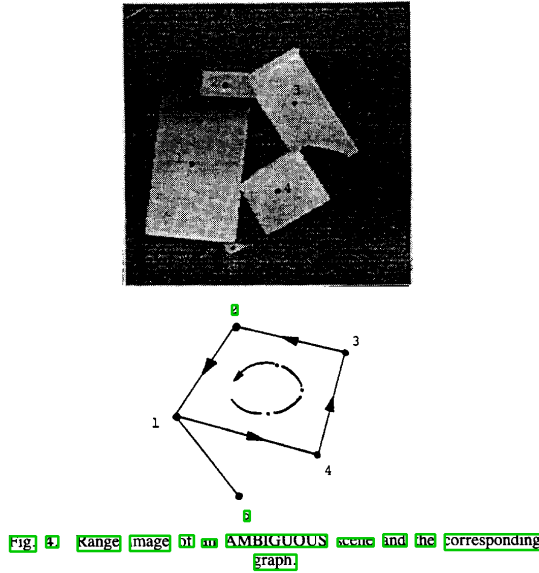


(e) Intersecção



(f) Refinamento

Figure 6: Segmentação de palavras utilizando operadores morfológicos.



sensory feedback is carried out in a local reflexive mode rather than in a planned mode with one exception, that is, when a pathological state is detected.

3) *States*: This is a finite set of states describing the environment of the Turing machine as perceived by the sensors. If new sensors are added, the set of states is partitioned to describe the scene as perceived by the additional sensors. For example, if a sensor capable of determining the "touch" relations of objects in the scene is added, then the set of five states, can be partitioned (a finer partition) to describe both the "touch" and "on-top-of" relations. The states of the machine are:

- Empty** If there are no vertices in the diagram, i.e., an empty diagram.
- Dispersed** If there are no edges in the diagram, i.e., a null diagram (Fig. 2).
- Overlapped** If there are at least two vertices connected with an edge (Fig. 3).
- Ambiguous** If there is one or more directed cycles in the diagram (Fig. 4).
- Unstable** This category is not tested by the analysis of the graph but through analysis of the contact point/line of the object with the support plane. If this contact is a point on a line, it is classified as unstable. See Fig. 5.

Figure 7: Demarcação (verde) dos blocos os quais englobam uma componente correspondente à uma palavra.

3.5 Limitações

Nas seções anteriores, vimos o seguinte processo: fechamento horizontal, fechamento vertical, inteseccção, refinamento e seleção. Destas etapas, temos que os fechamentos, o refinamento e a seleção exigem parâmetros adequados para que funcionem bem: os núcleos e *ratios* utilizados para palavras são diferente dos utilizados para linhas. Portanto, dependendo do objeto que queremos segmentar na imagem, estes parâmetros precisam ser reajustados. Outro detalhe importante, é que, como estes parâmetros são todos baseados em pixels, temos que tamanhos diferentes de palavras e textos também requerem que a resolução ou os parâmetros sejam reajustados, em outras palavras, os parâmetros não generalizam para diferentes tamanhos de texto e tipos de fonte. Podemos observar na **Figura 8** que, ao utilizarmos os mesmos parâmetros da **Figura 7**, apesar do método generalizar, os parâmetros devem ser regulados ou a imagem normalizada para que a segmentação seja feita de forma adequada.

The current font size is: 6pt
The current font size is: 10pt
The current font size is: 10pt
The current font size is: 10pt
The current font size is: 10.99pt
The current font size is: 12pt
The current font size is: 14.4pt
The current font size is: 17.28pt
The current font size is: 20.74pt
The current font size is: 24.88pt

(a) Tamanhos Diferentes

This paragraph is in a cursive font.
This paragraph is in a fantasy font.
This paragraph is in a sans-serif font.
This paragraph is in a serif font.

(b) Fonte Diferentes

Figure 8: Exemplos de textos diferentes utilizando os mesmos parâmetros.

4 Discussão e Conclusão

A partir do experimento, observamos a capacidade que operadores morfológicos apresentam de extrair significado da forma das imagens, no entanto, para que estes sirvam uma finalidade, é necessário que a imagem seja devidamente pré e pósprocessada para extrair sentido dos conjuntos transformados pelas operações. No caso da segmentação de texto, temos que o método aplicado é eficiente e apresenta boa acurácia em relação ao texto real, desde que os parâmetros estejam devidamente calibrados. A dificuldade que o método aplicado apresenta, é que as operações morfológicas estão atreladas aos pixels, portanto, quando temos diferentes tamanhos e formas de texto, dificilmente conseguimos utilizar os mesmos parâmetros para identificar linhas e palavras. De toda forma, operadores morfológicos são uma forma eficiente de extrair sentido das formas das imagens e também transformá-las (filtragem de ruído por exemplo).

References

- [1] R.C. Gonzalez. *Digital Image Processing*. Prentice Hall, 2007.